

Chapitre 2

Équations différentielles

ENSAM-Meknès

Université Moulay Ismail - Meknès

23 octobre 2011

- 1 Introduction
- 2 Méthode d'Euler explicite
- 3 Méthode de Taylor d'ordre 2
- 4 Méthodes de Runge-Kutta
- 5 Systèmes d'équations différentielles
- 6 Équations d'ordre supérieur
- 7 Méthodes à pas multiples
- 8 Etude générale de la convergence

On s'intéresse ici à des problèmes où l'inconnue est une fonction :

- Chute libre : Trouver la vitesse $v(t)$ satisfaisant

$$mv'(t) = mg - kv(t)^2 \quad \forall t$$

- Pendule : Trouver l'angle $\theta(t)$ satisfaisant

$$ml \theta''(t) = -mg \sin(\theta(t)) - lc_f \theta'(t) \quad \forall t$$

- Masse-ressort : Trouver le déplacement $x(t)$ satisfaisant

$$x''(t) + cx'(t) + \omega^2 x(t) = 0 \quad \forall t$$

- Croissance de population : Trouver le nombre d'individus $N(t)$

$$N'(t) = (n_1 - m_1 N(t))N(t) \quad \forall t$$

Equation différentielle

Une équation différentielle est une relation faisant intervenir les dérivées d'une fonction inconnue $y(t)$ à déterminer sur l'intervalle $[a, b]$:

$$F(t, y(t), y'(t), y''(t), \dots, y^{(n)}(t)) = 0 \quad \forall t \in [a, b]$$

Définition 1

L'ordre d'une équation différentielle est déterminé par d'ordre maximale de la dérivée de la fonction inconnue (variable dépendante).

Exemple

- Ordre 1 : $y'(t) = ty(t)$ alors $F(t, y, y') = y'(t) - ty(t)$
- Ordre 2 : $y''(x) = y'(x) - y(x)$ alors
 $F(x, y, y', y'') = y''(x) - y'(x) + y(x)$

- Contrairement à une équation algébrique, l'inconnue du problème est une fonction.
- La solution générale dépend de un ou plusieurs paramètres.
- Pour avoir unicité, il faut préciser des conditions
 - initiales (valeurs connues pour la plus petite valeur de la variable indépendante), généralement la variable indépendante est associée au temps (et notée t)

$$y'(t) = ty(t) \quad y(0) = 1$$

- des conditions aux limites, (valeurs connues à une/plusieurs extrémités de l'intervalle de la variable indépendante), généralement la variable indépendante est associée à une variable d'espace (et habituellement noté x) :

$$y''(x) = 2a \quad y(0) = c, \quad y(1) = a + b + c$$

Equation d'ordre 1

La formulation générale d'une équation différentielle d'ordre 1 avec condition initiale est :

$$\begin{cases} y'(t) &= f(t, y(t)) \\ y(t_0) &= y_0 \end{cases}$$

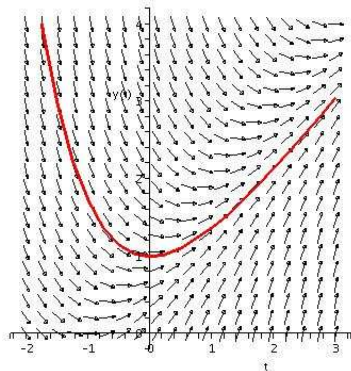
Exemple 1-2

- $y'(t) = t$ avec $y(0) = 1$ donnant $y(t) = \frac{t^2}{2} + 1$
- $y'(t) = ty(t)$ avec $y(3) = 42$ donnant $y(t) = 42e^{\frac{t^2-3^2}{2}}$
- En général il est impossible de résoudre analytiquement ces équations.
- Dans les cas simples on peut faire une analyse des directions de l'équation donnant une "image" de la solution

Exemple : Champs des directions de l'équation différentielle

$$\begin{cases} y'(t) &= -y + t + 1 \\ y(0) &= 1 \end{cases}$$

Dans le plan (t, y) on fait une analyse du “champ vectoriel” :
Pour chaque point (t, y) on peut calculer la dérivée de y .
Exemple en $(1, 1)$ on a $y'(1) = 1$ et en $(-1, 1)$ on a $y'(-1) = -1$
On peut alors “remplir” le plan et voir “apparaître” une solution



Rappels

- On s'intéresse aux méthodes numériques de résolution.
- Au niveau numérique, on ne peut pas évaluer la solution $y(t)$ en tout point t .
- On va chercher une approximation de la solution pour certaines valeurs de la variable indépendante t qu'on notera t_i .

Vocabulaire

- La distance entre deux valeurs successives t_{i+1} et t_i est noté $h_i = t_{i+1} - t_i$ s'appelle le **pas de temps**
- On utilisera généralement un pas de temps constant noté h
- On note $y(t_i)$ la solution analytique évaluée en $t = t_i$
- On note y_i la solution approximative en $t = t_i$ obtenue par une méthode numérique

Discrétisation

Supposons que l'on désire calculer la solution $y(t)$ d'une équation différentielle pour $t_0 \geq t$. Une **discrétisation** de l'équation définissant $y(t)$ est définie comme

- la donnée d'un ensemble discret de valeur pour la variable indépendante : en général, on fixe le pas h et on pose

$$t_i = t_0 + ih \quad \forall i = 0, 1, \dots, N$$

Produisant la suite

$$t_0 < t_1 < t_2 < \dots < t_i < \dots < Nh$$

- et d'une méthode numérique (schéma) fournissant une approximation de la solution exacte au points t_i :

$$y_i \approx y(t_i)$$

Soit le problème

$$\begin{cases} y'(t) &= f(t, y(t)) \\ y(t_0) &= y_0 \end{cases}$$

On se donne un pas de temps h et $t_i = t_0 + ih$.

Pour approximer $y(t_1) = y(t_0 + h)$ on propose d'utiliser l'information connue :

$$t_0, y(t_0) \text{ et } y'(t_0) = f(t_0, y(t_0)) = f(t_0, y_0)$$

On construit la droite $d_0(t)$ passant par (t_0, y_0) et de pente $f(t_0, y_0)$:

$$d_0(t) = y_0 + (t - t_0)f(t_0, y_0)$$

et on propose $y_1 = d_0(t_1) = y_0 + (t_1 - t_0)f(t_0, y_0) = y_0 + hf(t_0, y_0)$

Pour approximer $y(t_2) = y(t_0 + 2h)$ on utilise le même procédé. On construit une droite avec l'information connue en t_1 : la droite $d_1(t)$ passant par (t_1, y_1) et de pente $f(t_1, y_1)$:

$$d_1(t) = y_1 + (t - t_1)f(t_1, y_1)$$

et on propose $y_2 = d_1(t_2) = y_1 + (t_2 - t_1)f(t_1, y_1) = y_1 + hf(t_1, y_1)$

et on recommence au pas suivant donnant

$$y_{i+1} = y_i + hf(t_i, y_i) \quad i = 0, \dots$$

Définition

On qualifie d'**explicite** un schéma où l'évaluation de l'approximation ne nécessite pas de résolution d'un système non-linéaire. Plus simplement l'approximation s'exprime en fonction de valeurs connues uniquement.

On a

$$y'(t) = f(t, y(t)) \quad \forall t \in [t_0, Nh] \quad \Rightarrow \quad y'(t_i) = f(t_i, y(t_i)) \quad \forall i$$

On arrive à la même méthode en considérant l'approximation de la dérivée $y'(t_i)$ par la formule de différence avant

$$f(t_i, y_i) \approx y'(t_i) \approx \frac{y_{i+1} - y_i}{t_{i+1} - t_i} = \frac{y_{i+1} - y_i}{h}$$

menant à

$$\frac{y_{i+1} - y_i}{h} = f(t_i, y_i) \quad \forall i$$

Cette méthode est connue comme la méthode de Euler explicite.

Méthode d'Euler explicite

- Étant donné un pas de temps h , une condition initiale (t_0, y_0) et un nombre maximal d'itérations N
- $i = 0$
- 1 $y_{i+1} = y_i + hf(t_i, y_i)$
- 2 $t_{i+1} = t_i + h$
- 3 Écrire t_{i+1} et y_{i+1}
- 4 Si $i = N$ arrêt
- 5 $i = i + 1$ retour 1

Erreur

Il faut cependant noter qu'en général

$$y_i \neq y(t_i) \text{ et } y'(t_i) = f(t_i, y(t_i)) \neq f(t_i, y_i)$$

donc

$$d_i(t) \neq y(t_i) + (t - t_i)f(t_i, y(t_i))$$

Mais alors l'erreur commise dans l'approximation de y_i est introduite dans l'approximation de y_{i+1} et on a propagation de l'erreur !

La propagation de l'erreur n'est pas exclusive à la méthode de Euler explicite. Elle est typique des méthodes numériques pour les éq. diff. On aura propagation de l'erreur en passant d'une itération à l'autre, et de manière générale l'erreur d'approximation

$$|y(t_i) - y_i|$$

augmente (légèrement, quand tout va bien) avec i .

Soit l'équation différentielle :

$$\begin{cases} y'(t) &= -y(t) + t + 1, \\ y(0) &= 1. \end{cases}$$

qui admet la solution exacte $y(t) = e^{-t} + t$.

Le schéma d'Euler s'écrit

$$\begin{cases} y_0 &= 1, \\ y_{n+1} &= y_n + h(-y_n + t_n + 1). \end{cases}$$

Pour $h = 0.1$ on obtient

$$y_1 = y_0 + 0.1(-y_0 + t_0 + 1) = 1 + 0.1(-1 + 0 + 1) = 1$$

$$y_2 = y_1 + 0.1(-y_1 + t_1 + 1) = 1 + 0.1(-1 + 0.1 + 1) = 1.01$$

$$y_3 = y_2 + 0.1(-y_2 + 0.2 + 1) = 1.01 + 0.1(-1.01 + 0.2 + 1) = 1.029$$

...

Méthode d'Euler : $y'(t) = -y(t) + t + 1$

t_n	$y(t_n)$	y_n	$ y(t_n) - y_n $
0.0	1.000 000	1.000 000	0.000 000
0.1	1.004 837	1.000 000	0.004 837
0.2	1.018 731	1.010 000	0.008 731
0.3	1.040 818	1.029 000	0.011 818
0.4	1.070 302	1.056 100	0.014 220
0.5	1.106 531	1.090 490	0.016 041
0.6	1.148 812	1.131 441	0.017 371
0.7	1.196 585	1.178 297	0.018 288
0.8	1.249 329	1.230 467	0.018 862
0.9	1.306 570	1.287 420	0.019 150
1.0	1.367 879	1.348 678	0.019 201

Quel est l'effet de h sur l'erreur $|y(t_n) - y_n|$?

Définition 2

Une méthode de résolution est dite **méthode à un pas** si elle est de la forme

$$y_{n+1} = y_n + h\phi(t_n, y_n)$$

où ϕ est une fonction quelconque. La méthode est à un pas si elle exige uniquement la solution au temps précédent pour la résolution au temps courant.

Une méthode est à **pas multiples** si la solution au temps t_{n+1} exige la solution à plusieurs valeurs de temps précédent $t_n, t_{n-1}, t_{n-2}, \dots$

Définition 3

Erreur de troncature locale au temps t_n est

$$\tau_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - \phi(t_n, y(t_n))$$

Remarque

On utilise la solution exacte dans l'expression de l'erreur de troncature locale. C'est parce qu'on veut mesurer l'erreur introduite par l'utilisation du schéma numérique pour un pas donné en supposant la solution exacte aux itérations précédentes.

$$y(t_{n+1}) - y_{n+1} = y(t_{n+1}) - y_n - h\phi(t_n, y_n)$$

En supposant la solution exacte en t_n on a

$$y(t_{n+1}) - y_{n+1} = y(t_{n+1}) - y(t_n) - h\phi(t_n, y(t_n)) = h\tau_{n+1}(h)$$

Revenons à la méthode d'Euler explicite. Dans ce cas $\phi(t, y) = f(t, y)$. En utilisant Taylor (encore) autour de t_n on a

$$\begin{aligned}y(t_{n+1}) &= y(t_n + h) = y(t_n) + y'(t_n)h + \frac{y''(t_n)h^2}{2} + O(h^3) \\&= y(t_n) + f(t_n, y(t_n))h + \frac{y''(t_n)h^2}{2} + O(h^3)\end{aligned}$$

alors

$$\frac{y(t_{n+1}) - y(t_n)}{h} = f(t_n, y(t_n)) + \frac{y''(t_n)h}{2} + O(h^2)$$

L'erreur de troncature locale est donc :

$$\tau_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - f(t_n, y(t_n)) = \frac{y''(t_n)h}{2} + O(h^2)$$

C'est-à-dire

$$\tau_{n+1}(h) = O(h)$$

Conclusions

- Si on suppose que y_n correspond à la solution alors sur chaque intervalle $[t_n, t_{n+1}]$, l'erreur $E_n = |y(t_n) - y_n|$ sera de l'ordre de $O(h^2)$.
- Cependant par la propagation des erreurs on a au mieux une erreur correspondant à l'erreur de troncature locale, i.e. une erreur d'ordre $O(h)$

Remarques

- L'ordre de la troncature locale donne l'ordre de la méthode numérique, la méthode de Euler explicite est d'ordre 1. Attention de ne pas confondre avec l'ordre de l'équation différentielle.
- La méthode de Euler explicite est facile d'emploi mais peu précise. C'est pourquoi elle est peu utilisée.
- Pour réduire de moitié l'erreur dans la méthode de Euler explicite on doit diviser le pas de temps par 2 puisque la méthode est d'ordre 1.

Revenons à l'exemple 3 :

$$y'(t) = -y(t) + t + 1 \quad y(0) = 1 \quad y(t) = e^{-t} + t$$

En utilisant Euler explicite avec $h = 0.1$ et 0.05 on trouve

t_i	$h = 0.1$	erreur	$h = 0.05$	erreur
0.1	1.0	0.004837	1.007375	0.002537
0.2	1.01	0.008731	1.0237809	0.005050

On a bien une réduction de moitié de l'erreur pour une réduction de moitié du pas de temps.

Pour gagner en précision on peut "étendre" la méthode de Euler explicite en utilisant le développement de Taylor. Cette approche permet d'avoir une erreur de troncature locale d'ordre plus élevée.

On utilise le développement de Taylor pour la fonction $y(t)$

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + y'(t_n)h + \frac{y''(t_n)h^2}{2} + O(h^3)$$

Or

$$\begin{aligned} y''(t) &= \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} y'(t) \\ &= \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} f(t, y(t)) \end{aligned}$$

On obtient

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + hf(t_n, y(t_n)) \\ &+ \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) + O(h^3) \end{aligned}$$

En négligeant les termes d'ordre supérieurs à 2 en h , et en remarquant que le membre de droite est composé d'éléments connus (on peut évaluer les dérivées de f et elles sont évaluées à des points connus).

On obtient :

$$y(t_{n+1}) \approx y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right)$$

On “remplace” la solution exacte $y(t_n)$ par l'approximation y_n et on a le schéma :

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right)$$

Puisqu'on utilise y_n plutôt que la solution exacte on aura encore propagation d'erreur d'un pas de temps à l'autre.

Erreur de troncature locale

Suivant la notation on a

$$\phi(t, y) = f(t, y) + \frac{h}{2} \left(\frac{\partial f(t, y)}{\partial t} + \frac{\partial f(t, y)}{\partial y} f(t, y) \right)$$

Part construction (on peut facilement montrer que)

$$\tau_{n+1}(h) = O(h^2)$$

La méthode est d'ordre 2.

Schéma de Taylor d'ordre 2

- 1 Étant donné un pas de temps h , une condition initiale (t_0, y_0) et un nombre maximal d'itérations N
- 2 Pour $0 \leq n \leq N$:
- 3
$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right)$$
- 4 $t_{n+1} = t_n + h$
- 5 Écrire t_{n+1} et y_{n+1}
- 6 Arrêt

Méthode de Taylor : $y'(t) = -y(t) + t + 1$			
t_i	$y(t_i)$	y_i	$ y(t_i) - y_i $
0.0	1.000 000	1.000 000	0.000 000 (0.000 000)
0.1	1.004 837	1.005 000	0.000 163 (0.004 837)
0.2	1.018 731	1.019 025	0.000 294 (0.008 731)
0.3	1.040 818	1.041 218	0.000 400 (0.011 818)
0.4	1.070 302	1.070 802	0.000 482 (0.014 220)
0.5	1.106 531	1.107 075	0.000 544 (0.016 041)
0.6	1.148 812	1.149 404	0.000 592 (0.017 371)
0.7	1.196 585	1.197 210	0.000 625 (0.018 288)
0.8	1.249 329	1.249 975	0.000 646 (0.018 862)
0.9	1.306 570	1.307 228	0.000 658 (0.019 150)
1.0	1.367 879	1.368 541	0.000 662 (0.019 201)

- équation différentielle : unicité de la solution obtenue en ajoutant des conditions initiales ou au bords i.e. on se donne la solution en certains points.
- l'ordre de l'équation : déterminée par le degré de la plus haute dérivée de l'inconnue
- le pas de temps, discrétisation et schéma numérique, schéma explicite
- schéma d'Euler explicite :

$$y_{n+1} = y_n + hf(t_n, y_n)$$

- Dans les schémas pour les éq. diff. l'erreur se propage

- Schéma à un pas général :

$$y_{n+1} = y_n + h\phi(t_n, y_n)$$

- Erreur de troncature locale pour un schéma qui donne l'ordre du schéma.
- Pour augmenté l'ordre de l'erreur de troncature on produit la méthode de Taylor d'ordre 2 : vient du développmenet de Taylor de y autour du point t_n avec troncature des termes d'ordre 3 en h .

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right)$$

- Euler explicite est un schéma d'ordre 1
- La méthode de Taylor est d'ordre 2.

Revenons à $y'(t) = -y(t) + t + 1$. En utilisant Euler explicite avec $h = 0.1$ on a

$$|y(1.0) - y_{10}| = 0.019201$$

et avec la méthode de Taylor on a

$$|y(1.0) - y_{10}| = 0.000662$$

Avec le schéma d'Euler, quelle valeur de h faut-il pour avoir une erreur en $t = 1.0$ comparable à l'erreur due à la méthode de Taylor ?

Puisque Euler est un schéma d'ordre 1 on a

$$|y(1.0) - y_n| \approx Ch$$

On veut une erreur qui soit $0.019201/0.000662 \approx 30$ plus petite alors il faut prendre h à peu près 30 fois plus petit !

On devra donc prendre $\tilde{h} \approx 0.1/30$

Conclusions

- En prenant $\tilde{h} \approx 0.00333$ on aura une erreur comparable CEPENDANT on devra faire 300 itérations pour atteindre la valeur de $t = 1.0$ au lieu de 10.
- On peut donc contrôler l'erreur jusqu'à un certain point en modifiant le pas de temps mais c'est au détriment du nombre d'itérations.
- Le tableau p 355, procède au même traitement mais avec la méthode de Taylor d'ordre 2. En faisant une division par 2 du pas de temps on réduit par un facteur 4 l'erreur . On note qu'en fait le facteur 4 s'obtient quand h est très petit.
- Tout comme pour le schéma d'Euler le nombre d'itérations grandit (multiplié par 2 à chaque fois).

Le choix du pas de temps semble arbitraire. Qu'en est-il ?

Considérons le problème

$$\begin{cases} y'(t) &= -5y(t) \\ y(0) &= 1 \end{cases}$$

Dont la solution est $y(t) = e^{-5t}$. En utilisant Euler explicite avec un pas de temps h donné on a

$$y_0 = 1 \quad y_{n+1} = y_n - 5hy_n = y_n(1 - 5h) \quad n \geq 0$$

Mais alors

$$y_n = y_0(1 - 5h)^n = (1 - 5h)^n$$

Il est clair que dans le cas où $(1 - 5h) \leq 0$ la solution oscillera entre des valeurs positives et négatives :

$$h = 0.25 \Rightarrow y_n = (-0.25)^n \Rightarrow$$

$$y_0 = 1, y_1 = -0.25, y_2 = 0.0625, y_3 = -0.015625 \dots$$

Mais on sait déjà que la solution est strictement positive !

Conclusion

- Le schéma produira une approximation valable pour certaine valeur du pas de temps seulement !
- Dans le cas présent on veut $h < 0.2$.
- Puisqu'en partie la construction des schémas ainsi que l'erreur de troncature qui "qualifie" nos schémas est basée sur le développement de Taylor, la conclusion habituelle s'applique. Il est dangereux d'utiliser des points trop "loin" les uns des autres : il faut des pas de temps petits si on veut être prudent !

Puisque modifier le pas de temps n'est pas idéal pour améliorer la qualité de la solution, on regarde ce qui se passe en augmentant l'ordre de la méthode de Taylor (en tronquant "plus loin") :

Par exemple pour avoir un schéma d'ordre 3 :

$$\begin{aligned}
 y(t_{n+1}) = & y(t_n) + hf(t_n, y(t_n)) \\
 & + \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) \\
 & + \frac{h^3}{6} \left(\frac{\partial^2 f(t_n, y(t_n))}{\partial t^2} + 2 \frac{\partial^2 f(t_n, y(t_n))}{\partial t \partial y} f(t_n, y(t_n)) \right. \\
 & + \frac{\partial f(t_n, y(t_n))}{\partial t} \frac{\partial f(t_n, y(t_n))}{\partial y} + \frac{\partial^2 f(t_n, y(t_n))}{\partial y^2} (f(t_n, y(t_n)))^3 \\
 & \left. + \frac{\partial f(t_n, y(t_n))}{\partial y} \frac{\partial f(t_n, y(t_n))}{\partial y} (f(t_n, y(t_n)))^2 \right) + O(h^4)
 \end{aligned}$$

Donne un gain en précision au détriment de la simplicité de l'expression à évaluer.

On voudrait une/des méthodes d'ordres de plus en plus grand tout en évitant la complexité d'évaluations des méthodes de Taylor d'ordre élevée.

Dans les méthodes de Taylor on augmente l'ordre en "extrayant de l'information" des dérivées de la fonction f (c-à-d y'). C'est aussi la source des difficultés car on devra "complexifier" le schéma en évaluant des dérivée d'ordre de plus en plus élevé.

Pour avoir des schémas simples on évitera d'utiliser les dérivées de la fonction f . Pour garder l'ordre des méthode de Taylor sans utiliser les dérivées il nous faudra "prendre l'information ailleurs" mais il ne reste plus que f ! On devra donc faire des évaluations de f en plusieurs points.

C'est la prémisse des méthodes de Runge-Kutta (RK).

On s'intéresse d'abord à l'ordre 2.

La base de la méthode de Taylor d'ordre 2 est le développement :

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) + O(h^3)$$

On veut un schéma de même ordre de précision et ne faisant pas apparaître de dérivées. On propose comme base des schémas d'ordre 2 :

$$y(t_{n+1}) = y(t_n) + a_1 hf(t_n, y(t_n)) + a_2 hf(t_n + a_3 h, y(t_n) + a_4 h)$$

Où a_1, a_2, a_3 et a_4 sont à déterminer (pour satisfaire l'ordre 2).

Pour trouver ces coefficients on fait le développement de Taylor (cette fois en 2 variables) de la fonction f :

$$\begin{aligned}
 f(t_n + a_3 h, y(t_n) + a_4 h) &= f(t_n + v_0, y(t_n) + v_1) \\
 &= f(t_n, y(t_n)) + v_0 \frac{\partial f(t_n, y(t_n))}{\partial t} + v_1 \frac{\partial f(t_n, y(t_n))}{\partial y} + O(h^2) \\
 &= f(t_n, y(t_n)) + a_3 h \frac{\partial f(t_n, y(t_n))}{\partial t} + a_4 h \frac{\partial f(t_n, y(t_n))}{\partial y} + O(h^2)
 \end{aligned}$$

On remplace ensuite dans l'expression :

$$\begin{aligned}
 y(t_{n+1}) &= y(t_n) + (a_1 + a_2) h f(t_n, y(t_n)) \\
 &\quad + \frac{h^2}{2} \left(2a_2 a_3 \frac{\partial f(t_n, y(t_n))}{\partial t} + 2a_2 a_4 \frac{\partial f(t_n, y(t_n))}{\partial y} \right) + O(h^3)
 \end{aligned}$$

Il ne nous reste plus qu'à identifier les termes en les comparant aux termes utilisés pour la méthode de Taylor.

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2} \left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) + O(h^3)$$

$$y(t_{n+1}) = y(t_n) + (a_1 + a_2)hf(t_n, y(t_n)) + \frac{h^2}{2} \left(2a_2a_3 \frac{\partial f(t_n, y(t_n))}{\partial t} + 2a_2a_4 \frac{\partial f(t_n, y(t_n))}{\partial y} \right) + O(h^3)$$

Pour que les deux expressions concordent il faut :

$$a_1 + a_2 = 1$$

$$a_2a_3 = \frac{1}{2}$$

$$a_2a_4 = \frac{f(t_n, y(t_n))}{2}$$

$$y(t_{n+1}) = y(t_n) + a_1 h f(t_n, y(t_n)) + a_2 h f(t_n + a_3 h, y(t_n) + a_4 h)$$

$$a_1 + a_2 = 1, \quad a_2 a_3 = \frac{1}{2}, \quad a_2 a_4 = \frac{f(t_n, y(t_n))}{2}$$

Le système à résoudre est sous-déterminé (plus d'inconnues que de variables) on est donc assuré d'une solution et permet différentes variantes.

Deux variantes "classiques" menant à des schémas fréquemment rencontré :

- $a_1 = a_2 = \frac{1}{2}$, $a_3 = 1$ et $a_4 = f(t_n, y(t_n))$ mène au schéma d'**Euler modifiée** en remplaçant $y(t_n)$ par y_n et en négligeant les termes en $O(h^3)$ dans l'expression
- $a_1 = 0$, $a_2 = 1$, $a_3 = \frac{1}{2}$ et $a_4 = \frac{f(t_n, y(t_n))}{2}$ mène à la **méthode du point milieu** en remplaçant $y(t_n)$ par y_n et en négligeant les termes en $O(h^3)$ dans l'expression

Schéma RK2 : Euler modifiée

- On se donne h , (t_0, y_0) et un nombre maximal d'itérations N
- Pour $0 \leq n \leq N$:

① $\hat{y} = y_n + a_4 h = y_n + hf(t_n, y_n)$

② $y_{n+1} = y_n + a_1 hf(t_n, y_n) + a_2 hf(t_n + a_3 h, y_n + a_4 h)$
 $= y_n + \frac{h}{2} (f(t_n, y_n) + f(t_n + h, \hat{y}))$

③ $t_{n+1} = t_n + h$

④ Écrire t_{n+1} et y_{n+1}

⑤ $n = n + 1$ retour à 1

L'étape 1 correspond à la méthode de Euler explicite, d'où le nom de méthode d'Euler modifiée. La méthode est en fait composée de deux calculs : une prédiction par l'étape 1 suivie d'une correction (amélioration) de cette prédiction dans l'étape 2. C'est une méthode du type **prédicteur-correcteur**

Schéma RK2 : Méthode du point milieu

- Étant donné un pas de temps h , une condition initiale (t_0, y_0) et un nombre maximal d'itérations N
- Pour $0 \leq n \leq N$:

1 $k_1 = a_4 h = hf(t_n, y_n)$

2
$$y_{n+1} = y_n + a_1 hf(t_n, y_n) + a_2 hf(t_n + a_3 h, y_n + a_4 h)$$
$$= y_n + hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

3 $t_{n+1} = t_n + h$

4 Écrire t_{n+1} et y_{n+1}

5 $n = n + 1$ retour à 1

On évalue f au point $t_n + \frac{h}{2}$ d'où le nom de la méthode.

Le procédé de construction utilisé pour l'ordre 2 peut être appliqué pour les ordres suivants. Il faut noter que cette approche devient de plus en plus complexe.

Dans tous les cas on obtient un système sous-déterminé à résoudre pour les coefficients.

Le schéma vedette de la famille des RK est le schéma d'ordre 4 suivant :

$$y_{n+1} = y_n + \frac{1}{6} \left(\begin{aligned} &hf(t_n, y_n) + 2hf\left(t_n + \frac{h}{2}, y_n + hf(t_n, y_n)\right) \\ &+ 2hf\left(t_n + \frac{h}{2}, y_n + hf\left(t_n + \frac{h}{2}, y_n + hf(t_n, y_n)\right)\right) \\ &+ hf\left(t_n + h, y_n + hf\left(t_n + \frac{h}{2}, y_n + hf\left(t_n + \frac{h}{2}, y_n + hf(t_n, y_n)\right)\right)\right) \end{aligned} \right)$$

Ce schéma est tellement utilisé qu'il est connu sous le nom de "RK4" oubliant qu'en fait il y a une "famille" de méthodes de Runge-Kutta d'ordre 4.

Schéma RK4

- 1 Étant donné un pas de temps h , une condition initiale (t_0, y_0) et un nombre maximal d'itérations N
- 2 Pour $0 \leq n \leq N$:
- 3 $k_1 = hf(t_n, y_n)$
- 4 $k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$
- 5 $k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$
- 6 $k_4 = hf(t_n + h, y_n + k_3)$
- 7 $y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
- 8 $t_{n+1} = t_n + h$
- 9 Arrêt

Méthode de Runge-Kutta : $y'(t) = -y(t) + t + 1$			
t_i	$y(t_i)$	y_i	$ y(t_i) - y_i $
0.0	1.0	1.0	0.0
0.1	1.004 837 4180	1.004 837 5000	0.819×10^{-7}
0.2	1.018 730 7798	1.018 730 9014	0.148×10^{-6}
0.3	1.040 818 2207	1.040 818 4220	0.210×10^{-6}
0.4	1.070 320 0460	1.070 320 2889	0.242×10^{-6}
0.5	1.106 530 6597	1.106 530 9344	0.274×10^{-6}
0.6	1.148 811 6361	1.148 811 9343	0.298×10^{-6}
0.7	1.196 585 3034	1.196 585 6186	0.314×10^{-6}
0.8	1.249 328 9641	1.249 329 2897	0.325×10^{-6}
0.9	1.306 569 6598	1.306 579 9912	0.331×10^{-6}
1.0	1.367 879 4412	1.367 879 7744	0.333×10^{-6}

On semble avoir LA méthode.

Est-il possible qu'il soit avantageux d'utiliser une méthode moins précise (demandant moins d'effort de calculs à chaque itérations) avec un pas de temps plus petit plutôt qu'une méthode très précise (couteuse d'un point de vue calculatoire) et demandant moins d'itérations ?

On voudrait pourtant tenir compte de tout les facteurs :

- nombre d'itérations pour une valeur donnée de t_0
- quantité de calcul
- erreur sur l'approximation

On mesure le coût du calcul par le nombre total dévaluation de f

- RK4 demande 4 évaluations de f par itération
- RK2 demande 2 évaluations de f par itération
- Taylor d'ordre 2 1 évaluation de f , et de ces 2 dérivées partielles
- Euler explicite 1 évaluation de f

En utilisant $y' = -y + t + 1$ comme illustration on obtient en mesurant l'erreur en $y(1)$

Schéma	h	# de pas	# eval de f	erreur
Euler explicite	0.025	40	40	0.464×10^{-2}
RK2 Euler mod.	0.05	20	40	0.159×10^{-3}
RK4	0.1	10	40	0.333×10^{-6}

Conclusions :

- Il est toujours mieux d'utiliser la méthode dont l'ordre est le plus élevé possible
- On comprend pourquoi RK4 est la plus utilisée.

La forme générale d'un système de m équations différentielles avec conditions initiales s'écrit :

$$\begin{cases} y_1'(t) = f_1(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_1(t_0) = y_{1,0}) \\ y_2'(t) = f_2(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_2(t_0) = y_{2,0}) \\ y_3'(t) = f_3(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_3(t_0) = y_{3,0}) \\ \vdots & \vdots \\ y_m'(t) = f_m(t, y_1(t), y_2(t), \dots, y_m(t)) & (y_m(t_0) = y_{m,0}) \end{cases}$$

Ici encore, on note $y_i(t_n)$, la valeur exacte de la i^{e} variable dépendante en $t = t_n$ et $y_{i,n}$, son approximation numérique.

En général les équations sont **couplées** : il y a dépendance des inconnues entre elles.

Les m conditions initiales à droite nous garantisse l'unicité de la solution.

D'une manière compacte, on écrit un système sous la forme vectorielle

$$\frac{d\vec{y}}{dt}(t) = \vec{f}(t, \vec{y}(t))$$

- Toutes les méthodes s'étendent presque sans changement dans le cas vectoriel. Comme pour la résolution de système non linéaire on n'a qu'à faire un "passage" en dimension supérieure dans les schémas déjà construits.

Par exemple Euler explicite devient :

$$\vec{y}_{n+1} = \vec{y}_n + h\vec{f}(t_n, \vec{y}_n)$$

- La condition initiale devient vectorielle $\vec{y}(t_0) = \vec{y}_0$
- Les considérations sur la précision s'appliquent encore : RK4 est à favoriser.

La forme générale d'une équation différentielle d'ordre m avec conditions initiales est :

$$y^{(m)}(t) = f(t, y(t), y^{(1)}(t), y^{(2)}(t), \dots, y^{(m-1)}(t))$$

où $y^{(i)}(t)$ désigne la i^{e} dérivée de $y(t)$.

Pour avoir l'unicité il nous faut m conditions initiales (la solution comportera m constantes que l'on détermine de manière unique avec m valeurs connues (et indépendantes) de y). Les conditions initiales sont données par

$$y(t_0) = c_1, y^{(1)}(t_0) = c_2, \dots, y^{(m-1)}(t_0) = c_m.$$

On a des outils de résolution pour les systèmes d'équations différentielles d'ordre 1. On va transformer notre équation d'ordre m en m équations d'ordre 1.

Le principe de base consiste à écrire l'équation d'ordre supérieur sous la forme d'un système de m équations différentielles

$$\left\{ \begin{array}{ll} y_1'(t) &= y_2(t) \\ y_2'(t) &= y_3(t) \\ y_3'(t) &= y_4(t) \\ \vdots &\vdots \\ y_{m-1}'(t) &= y_m(t) \\ y_m'(t) &= f(t, y_1(t), y_2(t), \dots, y_m(t)) \end{array} \right. \quad \begin{array}{ll} y_1(t_0) &= c_1 \\ y_2(t_0) &= c_2 \\ y_3(t_0) &= c_3 \\ \vdots &\vdots \\ y_{m-1}(t_0) &= c_{m-1} \\ y_m(t_0) &= c_m \end{array}$$

on a bien

$$y_1^{(m)}(t) = y_2^{(m-1)}(t) = \dots y_m^{(1)}(t) = f(t, y_1(t), y_2(t), \dots, y_m(t))$$

Soit une équation différentielle :

$$y'(t) = f(t, y(t))$$

Le principe de ces méthodes consiste à intégrer l'équation dans l'intervalle $[t_n, t_{n+1}]$, c'est-à-dire :

$$\int_{t_n}^{t_{n+1}} y'(t) dt = \int_{t_n}^{t_{n+1}} f(t, y(t)) dt$$

ou encore :

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t)) dt$$

Cela nous amène à un algorithme de la forme :

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y(t)) dt$$

A partir d'une table de différences divisées, on remplace f

$$f(t, y(t)) \approx p_m(t)$$

par un polynôme d'interpolation.

Ainsi l'algorithme devient

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y(t)) dt \approx y_n + \int_{t_n}^{t_{n+1}} p_m(t) dt$$

Si on choisit les noeuds $t_n, t_{n-1}, t_{n-2}, \dots$, on obtient la famille des méthodes d'Adams-Bashforth.

Formules d'Adams-Bashforth

$$y_{n+1} = y_n + hf_n \quad (\text{ordre 1})$$

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1}) \quad (\text{ordre 2})$$

$$y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}) \quad (\text{ordre 3})$$

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \quad (\text{ordre 4})$$

- La méthode d'Adams d'ordre 1 correspond à la méthode d'Euler qui est un schéma à un pas.
- La méthode d'Adams d'ordre 2 est un schéma à 2 pas car dépend de y_n et de y_{n-1} .
- Pour démarrer un tel schéma, il faut connaître y_0 et y_1 . On peut utiliser une méthode à un pas du même ordre pour calculer y_1 (RK2).
- En général, la méthode d'Adams d'ordre m est un schéma à m pas.

- La méthode d'Adams d'ordre 1 correspond à la méthode d'Euler qui est un schéma à un pas.
- La méthode d'Adams d'ordre 2 est un schéma à 2 pas car dépend de y_n et de y_{n-1} .
- Pour démarrer un tel schéma, il faut connaître y_0 et y_1 . On peut utiliser une méthode à un pas du même ordre pour calculer y_1 (RK2).
- En général, la méthode d'Adams d'ordre m est un schéma à m pas.

- La méthode d'Adams d'ordre 1 correspond à la méthode d'Euler qui est un schéma à un pas.
- La méthode d'Adams d'ordre 2 est un schéma à 2 pas car dépend de y_n et de y_{n-1} .
- Pour démarrer un tel schéma, il faut connaître y_0 et y_1 . On peut utiliser une méthode à un pas du même ordre pour calculer y_1 (RK2).
- En général, la méthode d'Adams d'ordre m est un schéma à m pas.

- La méthode d'Adams d'ordre 1 correspond à la méthode d'Euler qui est un schéma à un pas.
- La méthode d'Adams d'ordre 2 est un schéma à 2 pas car dépend de y_n et de y_{n-1} .
- Pour démarrer un tel schéma, il faut connaître y_0 et y_1 . On peut utiliser une méthode à un pas du même ordre pour calculer y_1 (RK2).
- En général, la méthode d'Adams d'ordre m est un schéma à m pas.

Formules d'Adams-Moulton

$$y_{n+1} = y_n + hf_{n+1} \quad (\text{ordre 1})$$

$$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n) \quad (\text{ordre 2})$$

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1}) \quad (\text{ordre 3})$$

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \quad (\text{ordre 4})$$

- Ces méthodes sont basées sur les noeuds $t_{n+1}, t_n, t_{n-1}, t_{n-2}, \dots$
- A cause de la présence de y_{n+1} dans le membre de droite, le schéma devient implicite, i.e. exige la résolution d'une équation non linéaire pour calculer y_{n+1} .
- Pour contourner cette difficulté, on combine les formules d'Adams-Bashforth et d'Adams-Moulton en des schémas dits *prédicteurs-correcteurs*.

- Ces méthodes sont basées sur les noeuds $t_{n+1}, t_n, t_{n-1}, t_{n-2}, \dots$
- A cause de la présence de y_{n+1} dans le membre de droite, le schéma devient implicite, i.e. exige la résolution d'une équation non linéaire pour calculer y_{n+1} .
- Pour contourner cette difficulté, on combine les formules d'Adams-Bashforth et d'Adams-Moulton en des schémas dits *prédicteurs-correcteurs*.

- Ces méthodes sont basées sur les noeuds $t_{n+1}, t_n, t_{n-1}, t_{n-2}, \dots$
- A cause de la présence de y_{n+1} dans le membre de droite, le schéma devient implicite, i.e. exige la résolution d'une équation non linéaire pour calculer y_{n+1} .
- Pour contourner cette difficulté, on combine les formules d'Adams-Bashforth et d'Adams-Moulton en des schémas dits *prédicteurs-correcteurs*.

Schémas de prédiction-correction

$$y_{n+1}^p = y_n + hf_n$$

$$y_{n+1} = y_n + hf_{n+1}^p \quad (\text{ordre 1})$$

$$y_{n+1}^p = y_n + \frac{h}{2}(3f_n - f_{n-1})$$

$$y_{n+1} = y_n + \frac{h}{2}(f_{n+1}^p + f_n) \quad (\text{ordre 2})$$

Schémas de prédiction-correction (suite)

$$y_{n+1}^p = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2})$$

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1}^p + 8f_n - f_{n-1}) \quad (\text{ordre 3})$$

$$y_{n+1}^p = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1}^p + 19f_n - 5f_{n-1} + f_{n-2}) \quad (\text{ordre 4})$$

Schéma de prédiction-correction d'ordre 2

t	y_n^p	y_n	$ y(t_n) - y_n $
0.0	—	1.000 000 000	0
0.1	—	1.004 837 500	$0.819\,640 \times 10^{-7}$
0.2	1.019 111 875	1.018 640 031	$0.907\,218 \times 10^{-4}$
0.3	1.041 085 901	1.040 653 734	$0.164\,486 \times 10^{-3}$
0.4	1.070 487 675	1.070 096 664	$0.223\,381 \times 10^{-3}$
0.5	1.106 614 851	1.106 261 088	$0.269\,571 \times 10^{-3}$
0.6	1.148 826 758	1.148 506 695	$0.304\,940 \times 10^{-3}$
0.7	1.196 543 746	1.196 254 173	$0.331\,129 \times 10^{-3}$
0.8	1.249 241 382	1.248 979 396	$0.349\,568 \times 10^{-3}$
0.9	1.306 445 195	1.306 208 166	$0.361\,493 \times 10^{-3}$
1.0	1.367 725 911	1.367 511 462	$0.367\,979 \times 10^{-3}$

Rappels

Un schéma numérique à un pas est une équation de récurrence de la forme :

$$\begin{cases} y_{n+1} &= y_n + h\phi(t_n; y_n; h) \\ t_{n+1} &= t_n + h \end{cases}$$

L

'erreur de consistance au pas n est par définition l'erreur commise sur y_{n+1} , lorsqu'on prend pour les valeurs précédentes des y_k les valeurs exactes $z(t_k)$, ce qui donne la définition suivante pour les méthodes à un pas :

L'erreur de consistance est la suite

$$e_n = z(t_{n+1}) - z(t_n) - h\phi(t_n; z(t_n); h)$$